Access-protected data carrier

This invention relates to a data carrier having a semiconductor chip in which secret data are stored. The invention relates in particular to a smart card.

Data carriers containing chips are used in a great number of different applications, for example for performing monetary transactions, paying for goods or services, or as an identification means for access or admission controls. In all said applications the data carrier chip normally processes secret data which must be protected from access by unauthorized third parties. Said protection is ensured by, among other things, giving the inner structures of the chip very small dimensions so that it is very difficult to access said structures with the aim of spying out data processed in said structures. In order to impede access further, one can embed the chip in a very firmly adhering compound whose forcible removal destroys the semiconductor plate or at least the secret data stored therein. It is also possible to provide the semiconductor plate during its production with a protective layer which cannot be removed without destroying the semiconductor plate.

With corresponding technical equipment, which is extremely expensive but nevertheless fundamentally available, an attacker could possibly succeed in exposing and examining the inner structure of the chip. Exposure could be effected for example by special etching methods or a suitable grinding process. The thus exposed structures of the chip, such as conductive paths, could be contacted with microprobes or examined by other methods to determine the signal patterns in said structures. Subsequently, one could attempt to determine from the detected signals secret data of the data carrier, such as secret keys, in order to use them for purposes of manipulation. One could likewise attempt to selectively influence the signal patterns in the exposed structures via the microprobes.

The invention is based on the problem of protecting secret data present in the chip of a data carrier from unauthorized access.

This problem is solved by the feature combinations of claims 1 and 9.

The inventive solution does not aim, like the prior art, at preventing exposure of the internal structures of the chip and the mounting of microprobes. Instead

measures are taken to make it difficult for a potential attacker to infer secret information from any signal patterns intercepted. Said measures consist according to the invention in manipulating security-relevant operations so that the secret data used in performing said security-relevant operations cannot be determined without including further secret information. For this purpose the security-relevant operations are disguised or falsified with the aid of suitable functions before execution. In order to impede or even prevent in particular a statistical evaluation in case of multiple execution of the security-relevant operations, a random component enters into the disguising function. As a result, an attacker cannot determine the secret data from any data streams intercepted.

The security-relevant operation will be represented in the following by function h mapping input data x on output data y, i.e. $y = h(x)$. To prevent secret input data x from being spied out the invention provides for disguised function h_{R1R2} to be determined, so that the following holds:

$$y \otimes R_2 = h_{R1R2}\,(x \otimes R_1).$$

The security-relevant operation is now performed by means of disguised function h_{R1R2} whose input data are not authentic secret data x but disguised secret data $x \otimes R_1$ generated by combining authentic secret data x with random number R_1. Without knowledge of random number R_1 one cannot determine authentic secret data x from disguised secret data $x \otimes R_1$. As a result of applying disguised function h_{R1R2} to disguised secret data $x \otimes R_1$ one obtains disguised output data $y \otimes R_2$. From disguised output data $y \otimes R_2$ one can determine output data y by suitable combination. Before each new execution of the security-relevant function one can preset new random numbers R_1 and R_2 from which new disguised function h_{R1R2} is determined in each case. Alternatively, a plurality of disguised functions h_{R1R2} can be permanently stored, one of which is selected randomly before execution of the security-relevant operation. It is especially advantageous to use two functions h_{R1R2} and $h_{R1'R2'}$, random numbers R_1' and R_2' being the inverse values of random numbers R_1 and R_2 with respect to the type of combination selected for disguising. In a further variant, random numbers R_1 and R_2 can also be identical. In particular, random num-

bers R_1 and R_2 can be selected statistically independently so that there is no correlation between input and output data which can be used for an attack.

If further operations are executed before or after security-relevant operation h in question here, random numbers R_1 and R_2 can also be used for disguising the data processed with the further operations.

The inventive solution can be used especially advantageously for security-relevant operations containing nonlinear functions. With nonlinear functions one cannot apply known protective measures based on disguising the secret data before execution of the functions. Known protective measures presuppose that the functions are linear with respect to the disguising operations so that disguising can be undone after execution of the functions. In the inventive solution, however, not only the secret data are falsified or disguised but also the security-relevant operations processing the secret data. The disguising of the secret data and the security-relevant operations is coordinated such that the authentic secret data can be derived from the disguised secret data after execution of the security-relevant operations. Coordination between disguising of the secret data and the security-relevant operations can be realized especially simply if the security-relevant operations are realized in the form of tables, so-called lookup tables. In the stated tables each input value x has output value y associated therewith. The functions realized by the tables are executed by looking up output values y belonging to particular input values x.

The invention will be explained below with reference to the embodiments shown in the figures, in which:

Fig. 1 shows a smart card in a top view,

Fig. 2 shows a greatly enlarged detail of the chip of the smart card shown in Fig. 1 in a top view,

Figs. 3a, 3b, 3c and 3d show representations of lookup tables.

Fig. 1 shows smart card 1 as an example of the data carrier. Smart card 1 is composed of card body 2 and chip module 3 set in a specially provided gap in card body 2. Essential components of chip module 3 are contact surfaces 4 for producing an electric connection with an external device, and chip 5 electrically connected with contact surfaces 4. As an alternative or in addition to contact surfaces 4, a coil not

shown in Fig. 1 or other transfer means can be present for producing a communication link between chip 5 and an external device.

Fig. 2 shows a greatly enlarged detail of chip 5 from Fig. 1 in a top view. The special feature of Fig. 2 is that it shows the active surface of chip 5, i.e. it does not show all layers generally protecting the active layer of chip 5. In order to obtain information about the signal patterns in the interior of the chip one can for example contact exposed structures 6 with microprobes. Microprobes are very thin needles which are brought in electric contact with exposed structures 6, for example conductive paths, by means of a precision positioning device. The signal patterns picked up by the microprobes are processed with suitable measuring and evaluation devices with the aim of inferring secret data of the chip.

The invention makes it very difficult or even impossible for an attacker to gain access to in particular secret data of the chip even if he has managed to remove the protective layer of chip 5 without destroying the circuit and to contact exposed structures 6 of chip 5 with microprobes or intercept them in some other way. The invention is of course also effective if an attacker gains access to the signal patterns of chip 5 in another way.

Figures 3a, 3b, 3c and 3d show simple examples of lookup tables in which the input and output data each have a length of 2 bits. All table values are represented as binary data. The first line states input data x, and the second line output data y associated therewith in the particular column.

Figure 3a shows a lookup table for undisguised function h. Figure 3a indicates that input value $x = 00$ has output value $h(x) = 01$ associated therewith, input value 01 output value 11, input value 10 output value 10, and input value 11 output value 00. The lookup table according to Figure 3a represents nonlinear function h which is to be executed within the framework of a security-relevant operation. According to the invention, however, one does not use the lookup table shown in Figure 3a itself in executing the security-relevant operation, but derives a disguised lookup table from said lookup table according to Figures 3b, 3c and 3d.

Figure 3b shows an intermediate step in determining the disguised lookup table. The lookup table according to Figure 3b was generated from the lookup table

according to Figure 3a by EXORing each value of the first line of the table from Figure 3a with random number $R_1 = 11$. Thus, EXORing the value 00 of the first line and first column of the table from Figure 3a with the number 11 yields the value 11, which is now the element of the first line and first column of the table of Figure 3b. The remaining values of the first line of the table shown in Figure 3b are determined accordingly from the values of the first line of the table shown in Figure 3a and random number $R_1 = 11$. The table shown in Figure 3b could already be used as a disguised lookup table for processing secret data likewise disguised with random number $R_1 = 11$. The result would be the plaintext values to be read in line 2 of the table from Figure 3b.

One usually arranges the individual columns of a lookup table according to ascending input data x. A table determined by accordingly sorting the table in Figure 3b is shown in Figure 3c.

If the table according to Figure 3c is to be disguised further or yield as output values likewise disguised values rather than plaintext values, one applies a further EXOR operation with further random number R_2.

Figure 3d shows the result of applying said further EXOR operation. In said operation the elements of the second line of the table according to Figure 3c are each EXORed with random number $R_2 = 10$. The element in the second line and the first column of the table according to Figure 3d thus results from EXORing the element in the second line and first column of the table according to Figure 3c with random number $R_2 = 10$. The further elements of the second line of the table according to Figure 3d are formed accordingly. The first line of the table according to Figure 3d is adopted by Figure 3c unchanged.

With the table shown in Figure 3d one can determine likewise disguised output data from disguised input data. The thus determined disguised output data can be supplied to further operations for processing disguised data or one can determine plaintext data therefrom by EXORing with random number $R_2 = 10$.

Use of the table shown in Figure 3d makes it possible to perform nonlinear operations with disguised secret data and protect said secret data from unauthorized access. The security-relevant operations themselves are still also protected from un-

authorized access since differently disguised functions can be used at every execution of the operations and the security-relevant operations themselves cannot be inferred even if the disguised functions could be determined. After conversion to plaintext, however, both the original security-relevant operations and the operations performed with the aid of disguised functions yield identical results. For example, input value 00 yields output value 01 according to the table in Figure 3a. In order to check whether the disguised table shown in Figure 3d yields the same output value one must first EXOR input value 00 with random number $R_1 = 11$. As a result of said combination one obtains the value 11. According to the table from Figure 3d, input value 11 likewise yields output value 11. In order to determine the plaintext from said output value one must EXOR the output value with random number $R_2 = 10$. As a result of said combination one obtains the value 01 which exactly matches the value determined with the aid of the table shown in Figure 3a.

Disguising the security-relevant operations or the input values can be effected not only by EXORing but also by other suitable types of combination, for example modular addition. Furthermore, the invention is not limited to the application of nonlinear functions represented by means of lookup tables. One can also use any nonlinear and even linear functions for which a suitable disguised function can be determined.